BSTZ No. 042390P11058
Express Mail No. EL802874184US

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR PRESERVATION OF FAILURE STATE IN A READ DESTRUCTIVE MEMORY

Inventors

John I. Garney
Robert W. Faber
Rick Coulson

Prepared by:

Blakely, Sokoloff, Taylor & Zafman LLP
12400 Wilshire Boulevard, Suite 700
Los Angeles, California 90025
(714) 557-3800

METHOD AND APPARATUS FOR PRESERVATION OF FAILURE STATE IN A READ DESTRUCTIVE MEMORY

Field

The invention pertains generally to memory storage technology and devices. More particularly, the invention relates to a system for preserving a memory failure state detected during a read operation.

Background

Memory storage devices are widely used by electronic devices for storing and accessing data. These memory storage devices employ various storage media in various configurations to store information.

Memory storage devices may suffer failure conditions which corrupt or destroy stored data therein. For example hard faults may cause part or all of one or more memory storage locations to destroy or corrupt data stored therein. Hard faults may include manufacturing and design defects which cause one or more memory locations to permanently or intermittently corrupt or destroy stored data therein.

Memory storage failures may also occur as a result of soft faults. Soft faults include a number of environmental conditional which may cause permanent or intermittent data corruption in an otherwise good memory storage location. For example, soft faults may be caused by signal noise, electrical interference, temperature variations (i.e. heat), shock or any other perturbance in the storage media, device, or matrix which causes one or more stored data to change, become corrupt, or be destroyed.

When data is found to have been corrupted, some memory storage systems may force a crash to avoid using the corrupted data. That is, some memory storage technologies, such as volatile storage, will lose all of its data on a crash and the new data, after restarting, may not be corrupted if the original corruption was caused by a soft error or a hard error that can be detected during the restart process. However, other memory technologies, such as some nonvolatile memory storage, do not lose their data upon powering off or crashing the system. Because the corrupted data remains in the memory storage location, subsequent read operations are able to determine that the data is corrupt. This information may be utilized by the storage device and/or an application to determine if a particular memory location has suffered an uncorrectable, persistent and/or intermittent, memory storage failure.

In some memory storage technologies, an operation to read data from a memory location causes the data to be destroyed. This is often called a destructive read operation and may result from the type of storage media used or how the memory system is designed. Some nonvolatile memory storage devices for example have destructive read operations. Destruction of the data in a particular memory location may include erasing, clearing, resetting, and/or overwriting the memory location. In such memory devices, the data read must typically be written back after being read in order to behave in a nondestructive read memory device manner.

When stored data has been corrupted in a read destructive memory device, simply crashing before the data is written back would leave the memory location in an uninitialized state that could be incorrectly interpreted as an uncorrupted value upon restart. Since the data is destroyed by a destructive read

operation, the memory system loses the ability to determine whether a particular memory location has suffered an uncorrectable, persistent and/or intermittent, memory storage error. Alternatively, if an uncorrectable read was detected, writing the same uncorrected value back to the memory location may also cause a subsequent read to return data that is incorrectly interpreted as valid data. This may cause repeated corruption of data stored in that memory location.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram illustrating one embodiment of an electronic device where the failure state preservation aspect of the invention may be embodied.

Figure 2 is a block diagram illustrating one embodiment of an error correction code system as it may operate on a memory storage device.

Figure 3 is a block diagram illustrating one embodiment of a failure state preservation aspect of the invention as it may be employed with one embodiment of a destructive memory writeback system.

Figure 4 is a flow diagram illustrating one method of implementing the failure state preservation aspect of the invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of the invention, numerous specific details are set forth in order to provide a thorough understanding of the invention. However, one of ordinary skill in the art would recognize that the invention may be practiced without these specific details. In other instances well known methods, procedures, and/or components have not been described in detail so as not to unnecessarily obscure aspects of the invention.

One aspect of the invention provides a novel scheme to preserve the failure state of a memory location.

Figure 1 is a block diagram illustrating how the invention may be practiced in an electronic device 102. The electronic device 102 includes a storage device 104 to store data and a read/write controller 106 to manage access to the storage device. The controller 106 may serve to synchronize or sequence read and write operations to the storage device.

The storage device 104 may include various memory storage devices or media, including any number of temporary/volatile and/or permanent/non-volatile memory storage devices, in various configurations. According to one implementation, the memory storage device 104 may include read destructive memory storage media.

The read/write controller 106 may include any number of devices including a general purpose processor, an application specific processor, a programmable device, registers, and/or an application specific integrated circuit.

The invention may be embodied within a multi-component circuit, one or more integrated circuit devices, one or more computer instructions, and/or a machine-readable medium. For example, in Figure 1 the failure state preservation aspect of the invention may be embodied within the read/write controller 106, the storage device 104, and/or other devices or components internal or external to the electronic device 102.

As noted above, it is desirable to be able to determine when stored data has been uncorrectably corrupted or destroyed and, if so, to identify or tag the memory location(s) from where such data originated. Data is said to be corrupt or invalid when the data stored in the memory location or read from the memory location is different from the data as originally written to the memory location.

To determine when stored data has been corrupted or destroyed, memory storage systems may implement some form of error correction code (ECC). An error correction code typically uses an algorithm to append one or more correction bits or characters to the stored data. These correction bits or characters may be subsequently utilized when the stored data is read to determine if it has changed, i.e. the data has been corrupted or destroyed, since it was written. An ECC provides sufficient additional information to allow detection and correction of specific types and organizations of changes in the original data written. Additionally, some specific organizations of changes can be detected as uncorrectable.

Figure 2 is a block diagram illustrating one embodiment of an error correction code system as it may operate on a memory storage device. A control unit 202 synchronizes and manages access (i.e. read and write operations) to the memory storage device 206. The control unit 202 is coupled to a write logic block 204, which writes data to the memory storage

device 206, and to a read logic block 208, which reads data from the memory storage device 206.

The write logic block 204 receives data and writes it to the memory storage device 206 as synchronized by the control unit 202. The write logic block 204 may include an error correction code (ECC) encoder 214 to encode data to be written to the memory storage device 206. The ECC encoder serves to append bits or characters to the data so that subsequent data corruption may be identified and/or corrected by later read operations.

The read logic block 208 may access and/or retrieve data from the memory storage device 206 and provide it to other systems, devices, or processes as synchronized by the control unit 202. The read logic block 208 may include an ECC decoder 216 to decode data read from the memory storage device 206. The decoder serves to determine if the data read is corrupt, and if so, attempt to correct/reconstruct the data into its original form.

According to one implementation, illustrated in Figure 2, the memory storage device 206 may be a nonvolatile memory storage device with a destructive read mechanism. As noted above, with a destructive read memory, a read operation from a memory location destroys the data which was stored in that location. To prevent the loss of data, the data which is read may be written back (writeback data) into the same memory location in the storage device 206 from where it was read. In one implementation, the read logic block 208 may be coupled to the write logic block 204 to permit data to be written back into the memory storage device 206.

The read logic block 208 may read data from the nonvolatile destructive read memory storage device 206 causing

the data read to be destroyed from the memory location in where it had been stored. The read logic block 208 may then write the data back into the memory location from where it was read to prevent its loss.

According to one implementation, the write logic block 204 may be used by the read logic block 208 to write data back into the memory device. The write logic block 204 may include a channel select mechanism 212 to select from among a new data channel and a writeback data channel. The control unit 202 may synchronize and select from which data channel the write logic block 204 accepts data to write to the memory storage device 206. Thus, when the read logic block 208 reads data from the destructive read memory storage device 206, the control unit 202 may enable data from the writeback channel to be written back into the memory location from where it was read.

In one implementation, shown in Figure 2, the read logic block 208 sends decoded data to the write logic block 204 which in turn encodes it prior to writing the data in the storage memory device 206. In another implementation, the read logic block 208 may send encoded data to the write logic block 204 which simply writes the data in the storage memory device 206.

With nonvolatile destructive read memory, when stored data has been corrupted it would be advantageous to avoid writing the corrupted data back to the failed memory location. That is, continuing to write data to a failed memory location may cause repeated corruption of data stored in that memory location. In some instances, the data may later appear uncorrupted when in fact it is corrupt.

For example, in the system illustrated in Figure 2, even when the ECC decoder 216 determines that data from a particular memory location was uncorrectably corrupt, the writeback mechanism writes the data back into the memory location from where it was read. This destroys the fact that a particular memory location has suffered an uncorrectable failure.

Figure 3 is a block diagram illustrating one embodiment of the failure state preservation aspect of the invention as it may be employed with one embodiment of a destructive memory writeback system.

The failure state preservation aspect of the invention may be implemented in an error correction code and/or memory storage system similar to that illustrated in Figure 2.

According to one embodiment of the invention, a control unit 302 synchronizes and manages access (i.e. read and write operations) to the destructive read memory storage device 306. The control unit 302 is coupled to a write logic block 304, which writes data to the memory storage device 306, and to a read logic block 308, which reads data from the memory storage device 306.

As in the system illustrated in Figure 2, the read logic block 308 reads data from the memory storage device and utilizes a writeback mechanism to store the read data back into the memory location from where it was read. However, when the ECC decoder 316 detects that the read data is uncorrectably corrupted, instead of writing the data back into the memory location from where it was read, the read logic block 308 / ECC decoder 316 causes the failure state of the memory location to be preserved.

According to one embodiment, the failure state of the memory location is preserved by writing a codeword (Failure Codeword) into the memory location to indicate that the memory location has failed. A channel selector 310 may serve to permit the read logic block 308 to cause a failure codeword to be written to the memory storage device 306 directly rather than through the write logic block 304. In various other embodiments, the failure codeword may be written into the memory storage device 306 via the write logic block 304, either with or without going through the ECC encoder 314.

The failure codeword may be selected so that it is not a valid codeword used by the error correction code. The failure codeword may be any pattern or bits or characters uniquely selected to mark one or more failed memory locations. The failure codeword must be selected so that within the limits of the ECC correction ability it will not be interpreted as a read value that could result from some normal data write.

The failure codeword may also be chosen so that its mathematical distance is greater than all correctable data patterns. For example, a failure codeword may be chosen to have a mathematical distance from all correctable patterns of the ECC which is greater than the minimum distance of the ECC in use. This governs the probability that errors detected during subsequent read operations will cause the read data to appear as correctable data to the ECC decoder 316.

In one implementation, the failure codeword is a preselected value or pattern. In another implementation, the failure codeword may be dynamically adjusted based on the prior data read to maximize its mathematical distance from correctable data patterns. The value of the failure codeword may also be adjusted to meet specific application requirements

for failure detection and may be used to define an appropriate ECC for a given memory technology.

In one implementation, the memory location is marked as failed when the data corruption detected by the ECC decoder 316 is uncorrectable. That is, the corrupted data could not be recovered/corrected by the ECC decoder algorithm.

While nonvolatile destructive memory has been employed for purposes of illustration, the invention is not limited to this type of memory and may be implemented with other memory technologies. Similarly, although various components have been described in a particular configuration, the invention is not limited to the configuration(s) shown. For example, the write logic block 204 and 304 and read logic block 208 and 308 may not include an internal ECC encoder 214 and 314 or ECC decoder 216 and 316 respectively. That is, those components may be provided separate from the write logic block 204 and 304 and read logic block 208 and 308.

Figure 4 is a flow diagram illustrating one method of implementing the failure preservation system of the invention. At step 402, data is read from a memory location of a storage device (i.e. a read-destructive memory device). The read data is also corrected using the ECC. The data is examined to determine if it uncorrectably corrupted at step 404. This may be done in a number or ways known to those skilled in these arts. According to one implementation, error correction coding is used to ascertain if data has been corrupted. If the data is found to be valid (not corrupt) then it is written back into the memory location from which it was read at step 406. If the data is found to be corrupt, at step 408, a failure codeword is written into the memory location from which it was read to indicate a failure state for the particular memory location. As noted above, the failure

codeword may be preselected or dynamically chosen to uniquely mark a memory failure state. The failure codeword may be selected so that it appears as an uncorrectable error to an ECC decoder.

While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art. Additionally, it is possible to implement the invention or some of its features in hardware, programmable devices, firmware, software or a combination thereof where the software is provided in a processor readable storage medium such as a magnetic, optical, or semiconductor storage medium.